IN THE CLAIMS

Please amend claim 6 as indicated below.

This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

Claim 1 (original) A completion table, comprising:

a plurality of entries, wherein each of said plurality of entries tracks a consecutive number of outstanding instructions, wherein each of said plurality of entries is configured to store an instruction address of a first of said consecutive number of outstanding instructions and an identification of said first of said consecutive number of outstanding instructions.

Claim 2 (original) The completion table as recited in claim 1, wherein said consecutive number of outstanding instructions comprises a length of a cache line.

Claim 3 (original) The completion table as recited in claim 1, wherein said instruction address is an effective address.

Claim 4 (original) The completion table as recited in claim 1, wherein an instruction address and an identification of a next to complete instruction is calculated using said instruction address of said first of said consecutive number of outstanding instructions and said identification of said first of said consecutive number of outstanding instructions, respectively, in a selected entry of said completion table.

Claim 5 (original) The completion table as recited in claim 4, wherein said entry is selected based on one of an oldest active instruction and an instruction finished at an execution unit.

Claim 6 (currently amended) A method for tracking a larger number of outstanding instructions in a completion table comprising the steps of:

issuing instructions to a first and a second execution unit;

selecting an identification of one of [[an]] a [[instruction]] finished instruction and an [[instruction]] active instruction at one of said first and said second execution unit;

calculating an identification of a next to complete instruction using said selected identification of said ~~selected instruction~~ one of said finished instruction and said active instruction;

selecting an instruction address and an identification of a first of a consecutive number of outstanding instructions located in an entry of said completion table; and

calculating an instruction address of said next to complete instruction using said identification of said next to complete instruction and said selected instruction address and identification of said first of said consecutive number of outstanding instructions located in said entry of said completion table.

Claim 7 (original) The method as recited in claim 6 further comprising the step of:

calculating an address offset using said selected identification of said first of said consecutive number of outstanding instructions located in said entry of said completion table and said identification of said next to complete instruction.

Claim 8 (original) The method as recited in claim 7, wherein said selected instruction address and identification of said first of said consecutive number of outstanding instructions is selected from an entry located at a tail position of said completion table if said selected instruction is older than a first of a consecutive number of outstanding instructions located in an entry prior to said tail position of said completion table.

Claim 9 (original) The method as recited in claim 7, wherein said selected instruction address and identification of said first of said consecutive number of outstanding instructions is selected from an entry located prior to a tail position of said completion table if said selected instruction is not older than a first of a consecutive number of outstanding instructions located in said entry located prior to said tail position of said completion table.

Claim 10 (original) The method as recited in claim 7 further comprising the step of:

calculating an instruction address of said next to complete instruction using said calculated address offset and said selected instruction address.

Claim 11 (original) The method as recited in claim 10, wherein said instruction address of said next to complete instruction is calculated by adding said calculated address offset to said selected instruction address.

Claim 12 (original) The method as recited in claim 9 further comprising the step of:

deallocating an entry located at said tail position of said completion table if said selected instruction is not older than said first of said consecutive number of outstanding instructions located in said entry prior to said tail position of said completion table.

Claim 13 (original) The method as recited in claim 6, wherein said selected instruction is a most recently finished instruction at one of said first and said second execution unit.

Claim 14 (original) The method as recited in claim 13, wherein said identification of said next to complete instruction is calculated by adding a logical value of one to said identification of said selected instruction.

Claim 15 (original) The method as recited in claim 6, wherein said first execution unit is a floating point unit, wherein said second execution unit is a fixed point unit.

Claim 16 (original) The method as recited in claim 15, wherein said selected instruction is an instruction finished at said floating point unit if said selected instruction executed at said floating point unit and an instruction executed at said fixed point unit finish at a same time, wherein said identification of said next to

complete instruction is calculated by adding a logical value of one to said identification of said selected instruction.

Claim 17 (original) The method as recited in claim 15, wherein said selected instruction is an instruction finished at said fixed point unit if there is no instruction finishing at said floating point unit and said selected instruction is older than an oldest active instruction at said floating point unit, wherein said identification of said next to complete instruction is calculated by adding a logical value of one to said identification of said selected instruction.

Claim 18 (original) The method as recited in claim 15, wherein said selected instruction is an oldest active instruction at said floating point unit if there is no instruction finishing at said floating point unit and said selected instruction is younger than said oldest active instruction at said floating point unit, wherein a youngest finishing instruction at said fixed point unit is saved, wherein said identification of said next to complete instruction is said identification of said selected instruction.

Claim 19 (original) The method as recited in claim 18, wherein said selected instruction is said saved instruction if said oldest active instruction at said floating point unit is finished and there are no older instructions at said floating point unit than said finished oldest active instruction at said floating point unit and if there are no other younger instructions at said fixed point unit, wherein said identification of said next to complete instruction is calculated by adding a logical value of one to said identification of said selected instruction.

Claim 20 (original) The method as recited in claim 18, wherein said selected instruction is a youngest finishing instruction at said fixed point unit if said oldest active instruction at said floating point unit is finished and there are no older instructions at said floating point unit than said finished oldest active instruction at said floating point unit and if said youngest finishing instruction at said fixed point unit is younger than said saved instruction, wherein said identification of said next to

complete instruction is calculated by adding a logical value of one to said identification of said selected instruction.

Claim 21 (original) A processor, comprising:

an instruction fetch unit configured to fetch instructions;

an instruction dispatch unit coupled to said instruction fetch unit, wherein said instruction fetch unit is further configured to issue said fetched instructions to said instruction dispatch unit, wherein said instruction dispatch unit comprises an instruction queue configured to store said fetched instructions;

a first and a second execution unit coupled to said instruction dispatch unit, wherein said dispatch unit is configured to dispatch said stored fetched instructions to said first and said second execution unit; and

a completion unit coupled to said instruction fetch unit, wherein said instruction fetch unit is further configured to issue an instruction address and an identification of each of said fetched instructions to said completion unit, wherein said completion unit is configured to keep track of when said fetched instructions have been completed, wherein said completion unit comprises:

a completion table, wherein said completion table comprises a plurality of entries, wherein each of said plurality of entries tracks a consecutive number of outstanding instructions, wherein each of said plurality of entries is configured to store an instruction address of a first of said consecutive number of outstanding instructions and an identification of said first of said consecutive number of outstanding instructions.

Claim 22 (original) The processor as recited in claim 21, wherein said completion unit comprises:

logic for selecting an identification of one of an instruction finished and an instruction active at one of said first and said second execution unit;

logic for calculating an identification of a next to complete instruction using said identification of said selected instruction;

logic for selecting an instruction address and an identification of a first of a consecutive number of outstanding instructions located in an entry of said completion table; and

logic for calculating an instruction address of said next to complete instruction using said identification of said next to complete instruction and said selected instruction address and identification of said first of said consecutive number of outstanding instructions located in said entry of said completion table.

Claim 23 (original) The processor as recited in claim 22, wherein said completion unit further comprises:

logic for calculating an address offset using said selected identification of said first of said consecutive number of outstanding instructions located in said entry of said completion table and said identification of said next to complete instruction.

Claim 24 (original) The processor as recited in claim 23, wherein said selected instruction address and identification of said first of said consecutive number of outstanding instructions is selected from an entry located at a tail position of said completion table if said selected instruction is older than a first of a consecutive number of outstanding instructions located in an entry prior to said tail position of said completion table.

Claim 25 (original) The processor as recited in claim 23, wherein said selected instruction address and identification of said first of said consecutive number of outstanding instructions is selected from an entry located prior to a tail position of said completion table if said selected instruction is not older than a first of a consecutive number of outstanding instructions located in said entry located prior to said tail position of said completion table.

Claim 26 (original) The processor as recited in claim 23, wherein said completion unit further comprises:

logic for calculating an instruction address of said next to complete instruction using said calculated address offset and said selected instruction address.

Claim 27 (original) The processor as recited in claim 26, wherein said instruction address of said next to complete instruction is calculated by adding said calculated address offset to said selected instruction address.

Claim 28 (original) The processor as recited in claim 25, wherein said completion unit further comprises:

logic for deallocating an entry located at said tail position of said completion table if said selected instruction is not older than said first of said consecutive number of outstanding instructions located in said entry prior to said tail position of said completion table.

Claim 29 (original) The processor as recited in claim 22, wherein said selected instruction is a most recently finished instruction at one of said first and said second execution unit.

Claim 30 (original) The processor as recited in claim 29, wherein said identification of said next to complete instruction is calculated by adding a logical value of one to said identification of said selected instruction.

Claim 31 (original) The processor as recited in claim 22, wherein said first execution unit is a floating point unit, wherein said second execution unit is a fixed point unit.

Claim 32 (original) The processor as recited in claim 31, wherein said selected instruction is an instruction finished at said floating point unit if said selected instruction executed at said floating point unit and an instruction executed at said fixed point unit finish at a same time, wherein said identification of said next to complete instruction is calculated by adding a logical value of one to said identification of said selected instruction.

Claim 33 (original) The processor as recited in claim 31, wherein said selected instruction is an instruction finished at said fixed point unit if there is no instruction finishing at said floating point unit and said selected instruction is older than an oldest active instruction at said floating point unit, wherein said identification of said next to complete instruction is calculated by adding a logical value of one to said identification of said selected instruction.

Claim 34 (original) The processor as recited in claim 31, wherein said selected instruction is an oldest active instruction at said floating point unit if there is no instruction finishing at said floating point unit and said selected instruction is younger than said oldest active instruction at said floating point unit, wherein a youngest finishing instruction at said fixed point unit is saved, wherein said identification of said next to complete instruction is said identification of said selected instruction.

Claim 35 (original) The processor as recited in claim 34, wherein said selected instruction is said saved instruction if said oldest active instruction at said floating point unit is finished and there are no older instructions at said floating point unit than said finished oldest active instruction at said floating point unit and if there are no other younger instructions at said fixed point unit, wherein said identification of said next to complete instruction is calculated by adding a logical value of one to said identification of said selected instruction.

Claim 36 (original) The processor as recited in claim 34, wherein said selected instruction is a youngest finishing instruction at said fixed point unit if said oldest active instruction at said floating point unit is finished and there are no older instructions at said floating point unit than said finished oldest active instruction at said floating point unit and if said youngest finishing instruction at said fixed point unit is younger than said saved instruction, wherein said identification of said next to complete instruction is calculated by adding a logical value of one to said identification of said selected instruction.